Performance and Performance Engineering of the Community Earth System Model

P. H. Worley, A. P. Craig, J. M. Dennis, A. A. Mirin, M. A. Taylor, M. Vertenstein

April 14, 2011

Supercomputing 2011
Seattle, WA, United States
November 12, 2011 through November 18, 2011

Performance and Performance Engineering of the Community Earth System Model[*]

Patrick H. Worley
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN 37831-6173
worleyph@ornl.gov

Anthony P. Craig
National Center for Atmospheric Research
P.O. Box 3000
Boulder, CO 80307-3000
tcraig@ucar.edu

John M. Dennis
National Center for Atmospheric Research
P.O. Box 3000
Boulder, CO 80307-3000
dennis@ucar.edu

Arthur A. Mirin
Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA 94551
mirin@llnl.gov

Mark A. Taylor
Sandia National Laboratories
PO Box 5800
Albuquerque, NM 87185-0370
mataylo@sandia.gov

Mariana Vertenstein
National Center for Atmospheric Research
P.O. Box 3000
Boulder, CO 80307-3000
mvertens@ucar.edu

ABSTRACT

The Community Earth System Model (CESM), released in June 2010, incorporates new physical process and new numerical algorithm options, significantly enhancing simulation capabilities over its predecessor, the June 2004 release of the Community Climate System Model. CESM also includes enhanced performance tuning options and performance portability capabilities. This paper describes the performance engineering aspects of the CESM and reports performance and performance scaling on both the Cray XT5 and the IBM BG/P for four representative production simulations, varying both problem size and enabled physical processes. The paper also describes preliminary performance results for high resolution simulations using over 200,000 processor cores, indicating the promise of ongoing work in numerical algorithms and where further work is required.

Categories and Subject Descriptors

J.2 [**Physical Sciences and Engineering**]: Earth and atmospheric sciences; D.2.m [**Software Engineering**]: Miscellaneous—*performance engineering*

General Terms

Performance

[*]This work has been co-authored by contractors of the U.S. Government under contracts No. DE-AC05-00OR22725 and No. DE-AC52-07NA27344, and is released as LLNL Report LLNL-CONF-480472. Accordingly, the U.S. Government retains a nonexclusive, royalty free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

1. INTRODUCTION

The Community Earth System Model (CESM) is the latest in a series of climate models that have been developed by and maintained at the National Center for Atmospheric Research (NCAR), with contributions from external researchers funded by the U.S. Department of Energy (DOE), National Aeronautics and Space Administration and National Science Foundation [4]. CESM was released in June 2010. This was the first major public release in 6 years, since version 3 of the Community Climate System Model (CCSM3) in June 2004 [2]. In constrast to CCSM3, the CESM contains options for a terrestrial carbon cycle and dynamic vegetation, atmospheric chemistry and aerosol dynamics, and ocean ecosystems and biogeochemical coupling, all necessary for an earth system model, as distinct from a purely physical model like the CCSM3. (Version 4 of the CCSM (CCSM4) was released in April, 2010, but this was primarily a prerelease of the new CESM physical models and coupler infrastructure [12]. In particular, CESM is a superset of the CCSM4 in that it can be configured to run the same science scenarios as CCSM4.)

Investigating the impact of climate change is a computationally expensive process, requiring significant computational resources [22]. Climate is a statistical science and a single experiment requires multiple realizations, typically 5 to 10. However, making progress on this problem still requires achieving reasonable throughput rates for the individual realizations when integrating out to hundreds or thousands of simulation years. Climate models employ time-accurate numerical methods, and exploitation of significant parallelism in the time-direction has yet to be demonstrated in production climate models. For the CESM this leaves functional parallelism between the component models, parallelizing over the spatial dimensions, and loop-level parallelism exploited within a shared-memory multi-processor compute node or a single processor. The size of the spatial computational grids that can be used and still achieve the required throughput rates for long time integrations is small compared to other peta- and exa-scale computational science applications. As a consequence the maximum number of processors that can be applied in a typical (single realization) production run is also relatively small. Parallel algorithms need to be highly optimized for even a modest number of computational threads to make best use of the limited available parallelism.

High resolution exploratory science runs are also vital for validating model extensions and for preparing for next generation production problem scenarios and next generation computing architectures. Thus more traditional parallel scalability out to very large thread counts is equally important.

Finally, CESM is a community code that is evolving continually to evaluate and include new science. Thus it has been very important that the CESM be easy to maintain and port to new systems, and that CESM performance be easy to optimize for new systems or for changes in problem specification or processor count [10].

In this paper we give an overview of the performance engineering aspects of CESM, including description of performance optimization options and overall performance optimization methodology. We then describe performance results for four production-like simulations on the Cray XT5 system sited at Oak Ridge National Laboratory and the IBM BG/P system sited at Argonne National Laboratory. Finally we describe performance results for two experimental high resolution simulations, one using the current default numerical methods and one using a new, more scalable, numerical method for the atmosphere that is expected to become the default method in the near future.

2. CESM

CESM consists of a system of five geophysical component models: atmosphere, land, ocean, sea ice, and ice sheet. Two-dimensional boundary data (flux and state information) are exchanged periodically through a coupler component. The coupler coordinates the interaction and time evolution of the component models, and also serves to remap the boundary-exchange data in space [5]. The atmosphere model is CAM, the Community Atmosphere Model [3, 21]. The ocean model is POP, the Parallel Ocean Program [11, 25]. The land model is CLM, the Community Land Model [8, 23]. The sea ice model is CICE, the Community Ice Code [1, 15, 16]. The ice sheet model is CISM, the Community Ice Sheet Model [24].

CAM, POP, CLM, and CICE are all parallel models, supporting both distributed memory (MPI) and shared memory (OpenMP) parallelism. The parallel implementations are based for the most part on decompositions of the associated spatial domains. CISM, the most recent addition to CESM, is still a serial code and runs as a single process, but with an interface that allows it to run concurrently with the other components. The coupler is itself a parallel code, supporting MPI, but not OpenMP, parallelism.

The ocean model can run concurrently with all of the other geophysical component models. The sea ice and land models can run concurrently with respect to each other, but for science reasons they must run sequentially with respect to the atmosphere model. This limits the fraction of time when all CESM components can execute simultaneously. The coupler runs both sequentially between components and concurrently with components, depending on the work to be done.

The frequency of atmosphere-land coupling and atmosphere-sea ice coupling is relatively high, ranging from 48 to 96 times per simulation day for the simulations examined here. In contrast, the atmosphere and ocean models exchange data once per simulation day for the example production simulations and 4 times per day for the high resolution simulations. Currently the ice sheet model coupling is one-way, receiving surface data from the land once per simulated day and not returning anything.

In the first three versions of CCSM each component model and the coupler were run as separate executables assigned to nonoverlapping processor sets. As of CESM (and CCSM4), the entire system is now run as a single executable and there is greatly increased flexibility to select the component processor layout. It is typical for the atmosphere, land, and sea ice model to run on a common set of processors, while the ocean model runs concurrently on a disjoint set of processors. This is not a requirement, however, and CESM can now run with all components on disjoint processor subsets, all on the same processors, or any combination in between.

3. PERFORMANCE

Each component model has its own performance characteristics, and the coupling itself adds to the complexity of the performance characterization [9]. The first step in CESM performance optimization is to determine the optimized performance of each of the component models for a number of different processor counts for the given platform and problem specification. This performance information is then used to determine how to assign processors to components to maximize CESM throughput.

Each CESM component model is also a parallel application code in its own right and was developed for the most part independently from the other component models. In consequence, each has its own approaches to performance engineering. These are discussed in turn.

3.1 CAM

CAM is characterized by two computational phases: the dynamics, which advances the evolutionary equations for the atmospheric flow, and the physics, which approximates subgrid phenomena such as precipitation processes, clouds, long- and short-wave radiation, and turbulent mixing. Separate data structures and parallelization strategies are used for the dynamics and physics. The dynamics and physics are executed in turn during each model simulation timestep, requiring that some data be rearranged between the dynamics and physics data structures each timestep.

CAM includes multiple compile-time options for computing the dynamics, referred to as dynamical cores or dycores. The default dycore for use with CESM is a finite-volume method (FV) formulated originally by Lin and Rood [18] that uses a tensor-product longitude $\times$ latitude $\times$ vertical-level computational grid over the sphere. CAM also supports less structured but more uniform grids such as cubed-sphere and icosahedral-like tesselations of the sphere and several dycores which can use these grids are being evaluated in CAM.

The first of these to reach sufficient maturity for inclusion in the CESM uses the spectral element (SE) method on a cubed-sphere grid [7]. The SE dycore is included in our high resolution performance experiments.

The parallel implementation of the FV dycore is based on two-dimensional tensor-product "block" decompositions of the computational grid into a set of geographically contiguous subdomains. A latitude-vertical decomposition is used for the main dynamical algorithms and a longitude-latitude decomposition is used for a Lagrangian surface remapping of the vertical coordinates and (optionally) geopotential calculation. Halo updates are the primary MPI communications required by computation for a given decomposition. OpenMP is used for additional loop-level parallelism.

CAM physics is based on vertical columns, and dependencies occur only in the vertical direction. Thus computations are independent between columns. The parallel implementation of the physics is based on the assignment of columns to (MPI) processes, and then to (OpenMP) threads within a process, representing a fine-grain longitude-latitude decomposition.

Transitioning from one grid decomposition to another, for example, latitude-vertical to longitude-latitude or dynamics to physics, may require that information be exchanged between processes. If the decompositions are very different, then every process may need to exchange data with every other process. If they are similar, each process may need to communicate with only a small number of other processes (or possibly none at all).

The computational cost in the physics is not uniform over the vertical columns, with the cost for an individual column depending on both geographic location and on simulation time. A number of predefined physics decompositions are provided that attempt to minimize the combined effect of load imbalance and the communication cost of mapping to/from the dynamics decompositions.

Common performance optimization options include [19, 20, 26]:

- the number of OpenMP threads per process;
- the number of processes to use in the dynamics latitude-vertical decomposition, in the dynamics longitude-latitude decomposition, and in the physics longitude-latitude decomposition (these need not be the same);
- for a given process count, the two-dimensional virtual processor grid used to define a dynamics decomposition;
- the physics load balancing option (and decomposition); and
- the MPI communication algorithms and protocols used for each communication phase, e.g. halo update or potentially nonlocal communication operators for mapping between decompositions.

The number of tuning options is relatively large for CAM, but reasonable defaults have been identified for common science scenarios, grid sizes, thread counts, and target architectures. Further optimization begins with these defaults.

3.2 CLM

CLM is a single column (snow-soil-vegetation) model of the land surface, and in this aspect it is embarassingly parallel. When using the FV dycore in the atmosphere model, CLM typically uses the same horizontal computational grid as the atmosphere. However, CESM supports the option of CLM using a totally different grid.

Spatial land surface heterogeneity in CLM is represented as a nested subgrid hierarchy in which grid cells are composed of multiple landunits, landunits are composed of multiple snow/soil columns, and snow/soil columns are composed of multiple plant functional types [13, 14]. Grid cells are grouped into blocks of nearly equal computational cost, and these blocks are subsequently assigned to MPI processes. When run with MPI-only parallelism, each process has only one block. When OpenMP is enabled, the number of blocks per process is by default set to the maximum number of OpenMP threads available. This number can be overridden at runtime.

A single load balancing algorithm is supported that has proven to work well across a variety of computer architectures and problem specifications. The common performance optimization options are:

- the number of OpenMP threads per process;
- the number of grid cell blocks assigned to each process

The default of one block per computational thread is typically optimal. Moreover, for a fixed core count, MPI-only often outperforms hybrid MPI/OpenMP runs. As described later, support for OpenMP is still important when optimizing performance of CESM as a whole.

3.3 POP

POP approximates the three-dimensional primitive equations for fluid motions on a generalized orthogonal computational grid on the sphere. Each timestep of the model is split into two phases. A three-dimensional "baroclinic" phase uses an explicit time integration method. A "barotropic" phase includes an implicit solution of the two-dimensional surface pressure using a preconditioned conjugate gradient solver.

The parallel implementation is based on a two-dimensional tensor-product "block" decomposition of the horizontal dimensions of the three-dimensional computational grid. The vertical dimension is not decomposed. The amount of work associated with a block is proportional to the number of grid cells located in the ocean. Grid cells located over land are "masked" and eliminated from the computational loops. OpenMP parallelism is applied to loops over blocks assigned to an MPI process. If specified at compile time, the number of MPI processes and OpenMP threads will be used to choose block sizes such that enough blocks are generated for all computational threads to be assigned work. The block sizes can also be specified manually.

The parallel implementation of the baroclinic phase requires only limited nearest-neighbor MPI communication (for halo upates) and performance is dominated primarily by computation. The barotropic phase requires both halo updates and global sums (implemented with local sums and a call to MPI_Allreduce for a small number of scalars) for each iteration of the conjugate gradient algorithm. The parallel performance of the barotropic phase is dominated by the communication cost of the halo updates and global sum operations [17],

Two different approaches to domain decomposition are considered here: "cartesian" and "spacecurve" [25]. The cartesian option decomposes the grid onto a two-dimensional virtual processor grid, and then further subdivides the local subgrids into blocks to provide work for OpenMP threads. The spacecurve option begins by eliminating blocks having only "land" grid cells. A space-filling curve ordering of the remaining blocks is then calculated, and an equipartition of this one-dimensional ordering of the blocks is used to assign blocks to processes.

The common performance optimization options are:

- the number of OpenMP threads per process;
- cartesian or spacecurve decomposition strategy; and
- the block size

3.4 CICE

The CICE sea ice model is formulated on a two-dimensional horizontal grid representing the earth's surface, typically using the same horizontal grid as POP. An orthogonal vertical dimension exists to represent the sea ice thickness. Similar to POP, the parallel implementation decomposes the horizontal dimensions into two-dimensional blocks. The vertical dimension is not decomposed. CICE exploits MPI and OpenMP parallelism over the same dimension, namely grid blocks. Currently the CICE decomposition is static and set at initialization. Like POP, a block size will be picked based on the total number of computational threads, or a block size can be specified manually.

The relative cost of computing on the sea ice grid varies significantly both spatially and temporally over a climate simulation because the sea ice distribution is changing constantly. This has a huge impact on the load balance of the sea ice model in a statically decomposed model. In general the load balance will be optimized if grid cells from varied geographical locations are assigned to each process. CICE also performs regular and frequent halo updates with a resultant performance cost that also depends on the assignment of grid cells to processes.

Optimal static load balance is achieved by balancing the computational load imbalance and the communication cost of halo updates. In this study we consider two decomposition algorithms: a simple two-dimensional cartesian decomposition that groups together strips of neighboring gridcells that span relatively large swaths of latitude, and a space-filling curve approach similar to that used in POP but with weights to identify where sea ice is more likely to occur [16]. We will again refer to these as "cartesian" and "spacecurve", respectively. If information is known about the likely distribution of sea ice, perhaps from an earlier simulation, this information can be exploited to improve the performance of the space-filling curve approach.

The common performance optimization options are:

- the number of OpenMP threads per process;
- cartesian or spacecurve decomposition strategy; and
- the block size

3.5 CISM

In current CESM configurations CISM uses a low resolution computational grid, and the computational cost of CISM is handled easily with one process. In this study CISM plays little role in determining model performance. This will change in the near future when more accurate, but more expensive, model formulations are introduced and when high resolution grids and two-way, higher frequency coupling are used. A parallel implementation of CISM is being developed to handle this increased computational cost.

3.6 Coupler

The CESM coupler is responsible for several actions, including rearranging data between different process sets, interpolating (mapping) data between different grids, merging data from different components, flux calculations, and diagnostics. Many of the algorithms are trivially parallel and require no communication between grid cells.

The coupler receives grid information in parallel at runtime from all of the model components. Domain decompositions are determined on the fly based upon the model resolutions, the component model decompositions, and the processors used by the coupler. Both rearrangement and mapping require interprocess communication, and the choice of MPI communication algorithm and protocol used to implement these affect performance. Performance is primarily determined by the number of processes assigned to the coupler, but the process counts and the placements of these components relative to the coupler processes can also have an impact. The coupler is the one component that can not reliably be optimized separately from the full CESM.

To summarize, the performance optimization options are:

- MPI communication algorithms and protocols used in tranferring data to and from the geophysical components; and
- number and layout of processes used for each component.

Note that OpenMP parallelism has been introduced in a development version of the coupler, but has not yet been included in a CESM release and was not used in these studies. As performance of the other components benefited from exploiting OpenMP parallelism on the multi-core node architectures of the target systems, we expect similar performance improvements in the coupler. With the possible exception of the largest simulation, coupler performance was not however a limiting factor in the current study. The qualitiative results described here should be unchanged after OpenMP parallelism is implemented in the coupler.

3.7 Parallel I/O

I/O is required for all components, and an efficient parallel I/O subsystem is critical. To address this need for CESM, a parallel I/O library called PIO has been developed and included in the release [6]. Each component specifies the number and location of I/O processes (starting process and stride between processes using the component process ordering) and what "standard" I/O library to use (currently netcdf or pnetcdf). PIO takes care of rearranging data to/from the I/O processes.

Primary performance optimization options include

- for each component, number and layout of I/O processes and underlying I/O library; and
- MPI communication algorithms and protocols used in communication to/from the I/O processes.

3.8 CESM

The methodology for optimizing CESM performance is as follows. For each problem specification and target architecture, generate a scaling curve or table of optimized performance for each component (for some appropriate range of computational threads). Based on an upper bound on the total number of processor cores or on an upper bound on total runtime, then choose a processor core count for each component and a layout that should satisfy the requirements. If successful, then try variants of this feasible configuration, to see if performance can be further improved without increasing the number of processor cores used, or to see if performance can be maintained while decreasing the processor core count. If unsuccessful, more conservative initial configurations are examined until a feasible configuration is identified, if possible. As part of these empirical investigations a resonable processor count and layout for the coupler must also be determined.

Based on the timing dependencies between the models discussed earlier, we typically start with a configuration where the ocean and the other components are assigned disjoint sets of processors, where the sea ice and land are assigned disjoint sets of processors, and where the atmosphere and sea ice and the atmosphere and land share processors. However, there can be problems with this layout on the current architectures. If components share multi-core processors, they typically must use the same number of OpenMP threads per processor or one of the components will leave processor cores idle when running. This is the reason that OpenMP is useful in CLM even when MPI-only is more efficient for a given core count. When sharing multi-core processors with the atmosphere, for which OpenMP can be very important, processor cores would be idle when the land was running if OpenMP threads were not used. Even if not as efficient as MPI-only on all of the cores, the land is still faster when using OpenMP in this situation than it would be if cores were left idle.

Other problems can arise for very large problem sizes, where components can not share processors because of memory requirements. While the resulting fully concurrent layouts, i.e. components not sharing processors, are inefficient, they still allow the problem to be solved.

The above methodology will be demonstrated in the results sections 5 and 6

4. TARGET PLATFORMS

CESM performance was evaluated on two Leadership Computing Facility (LCF) platforms, the Oak Ridge LCF Cray XT5 JaguarPF system and the Argonne LCF IBM Blue Gene/P (BG/P) Intrepid system.

Each of the 18,688 compute nodes in the XT5 system consists of two hex-core AMD Opteron 2435 (Istanbul) processors running at 2.6 GHz, for a total of 224,256 cores. Each compute node also has 16GB of DDR2-800, and nodes are connected with Cray SeaStar 2+ routers in a three-dimensional torus geometry.

Each of the 40,960 nodes in the BG/P system consists of a quad-core processor, for a total of 163,840 cores. Each core is 850 MHz PowerPC 450 32-bit microprocessor with a 64-bit dual-pipe floating point multiply-add unit. Compute nodes are connected via six networks, four of which are of importance to user applications: a three-dimensional torus, a global collective network, a global barrier and interrupt network, and 10 Gigabit Ethernet (between I/O nodes).

The experiments described in Section 5 took place in September and October, 2010. The first set of experiments in Section 6 (for simulation IIIa) took place during the summer of 2010. The second and third experiments in Section 6 (for the dycore comparison and for simulation IVa) took place during February and March of 2011.

5. PRODUCTION RESOLUTION SIMULATIONS

We begin with 4 relatively small problems, representing typical production simulations (for a single realization). We examine performance for two different computational grid resolutions for the atmosphere and land:

I) 1.9×2.5 degree resolution horizontal grid (144×96)
II) 0.94×1.25 degree resolution horizontal grid (288×192)

both coupled with ocean and sea ice using a 1 degree resolution horizontal computational grid (384×320) and 60 vertical levels for the ocean.

For each of these two grid resolutions we examine performance for two problem scenarios:

a) B1850_CN: all active components, pre-industrial, with CN (Carbon Nitrogen) in CLM
b) B1850_CAM5: all active components, pre-industrial, "CAM5" physics

We will not attempt to describe the scientific differences between these two problems here. The significant performance-related differences are as follows.

- The atmosphere computational grid has 26 vertical levels in B1850_CN and 30 vertical levels in B1850_CAM5, affecting the cost of both CAM dynamics and CAM physics.
- The number of tracers advected in the atmosphere dynamics and communicated between the physics and the dynamics during any data rearrangement is 3 in B1850_CN and 25 in B1850_CAM5.
- In B1850_CAM5 as compared to B1850_CN, a number of physical processes are computed more accurately and new processes are represented. The computational cost of column physics in B1850_CAM5 is 6-7 times greater than that in B1850_CN on both the XT5 and BG/P.

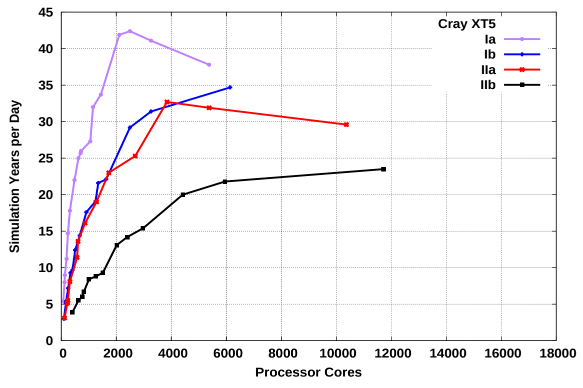


Figure 1: Best observed CESM performance for simulations Ia, Ib, IIa, IIb on the XT5

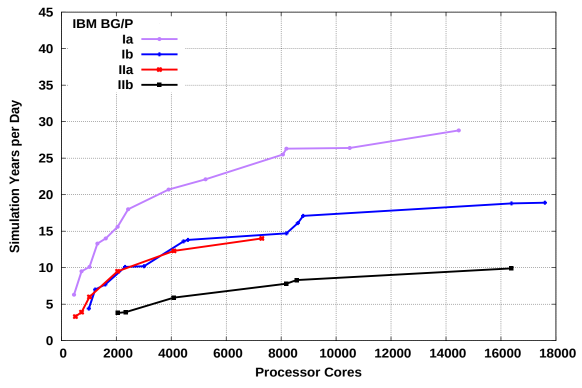


Figure 2: Best observed CESM performance for simulations Ia, Ib, IIa, IIb on the BG/P

- Including the Carbon-Nitrogen cycle (B1850_CN) increases the cost of the land model by approximately 40% on the XT5 and by 25% on the BG/P as compared to B1850_CAM5.

Using the above enumerations, we refer to the 4 simulations as Ia, Ib, IIa, and IIb.

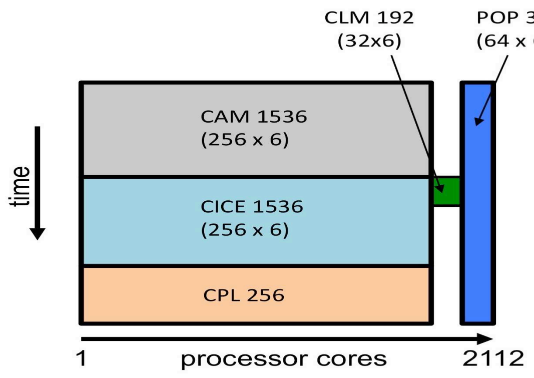
Best observed performance for the four benchmark problems for a range of processor core counts on the XT5 and BG/P is portrayed in Figures 1 and 2, respectively. The metric is a throughput rate, so larger is better. From these data a few immediate results are obvious.

- The (7 times) more expensive CAM5 physics, residing as it does in the embarrassingly parallel atmospheric physics, degrades performance by less than a factor of 2 at large processor core counts. The performance difference would be even lower if the increase in the number of tracers and number of vertical levels did not increase the cost of the atmospheric dynamics as well.
- The low latency and fast global reductions that characterize the BG/P (as compared to the XT5) allows increased performance scalability. However, this primarily delays or eliminates performance "rollover" rather than achieving any significant absolute performance improvement. In particular, the BG/P performance never catches up with that on the XT5 for these benchmark problems.

Figure 3 describes individual component scaling for the Ia benchmark on the XT5 in terms of average number of seconds per simulation day. A log-log scale is used because of the large range in the data being plotted. Here the ocean is the most expensive single component. However, what matters is the cost of the ocean relative to the combined times for the atmosphere, the sea ice, and the coupler. Because of this, the ocean is not necessarily the performance limiter at scale. Note that in Fig. 3 the data for a given processor core count is the minimum over all experiments that use this number of cores, including possibly MPI-only or 1, 2, 3, or 6 OpenMP threads per process. Also, data for process core counts for which performance is worse than that for smaller processor core counts are not plotted except for the largest core counts. Because nonoptimal processor core counts or OpenMP thread counts per process for a given component may be useful when optimizing CESM performance, these "omitted" data are still important in practice.

Figure 6 is a graph of the time spent in each geophysical component for the CESM performance data for simulation Ia in Fig. 1. Note that the x-axis is different from that in Fig. 3. When optimizing CESM performance, the resources allocated to a component may not increase monotonically as a function of total CESM process core count. The goal is to balance the ocean cost with that of the sum of atmosphere, sea ice, and coupler costs. (Meaningful timing data for the coupler is difficult to obtain directly and is omitted here.).

The following cartoon depicts the layout for the 2112 processor core count configuration for simulation Ia on the Cray XT5. The number of cores used is listed per component, as well as the number of MPI tasks and number of OpenMP threads per task (`tasks` × `threads`). The time direction indicates the relative amount of time spent in each component, and whether components run sequentially or concurrently with each other. Note that the coupler will sometimes run sequentially with the ocean, for example when the ocean and atmosphere are communicating, and sometimes concurrently. The amount of time indicated for the coupler includes time in which the other components are waiting for the ocean.



For comparison, Figures 4 and 7 are component performance scaling for problem Ia on the BG/P and time spent in each

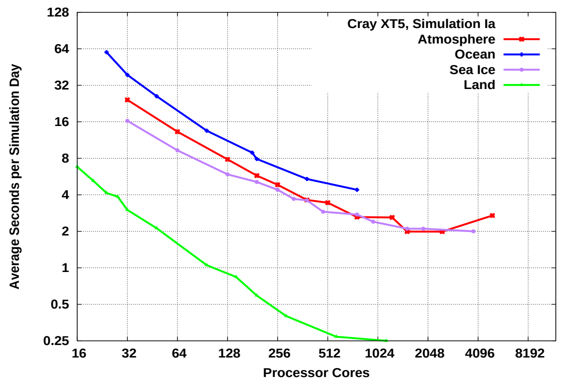


Figure 3: Best observed component performance for simulation Ia on the XT5

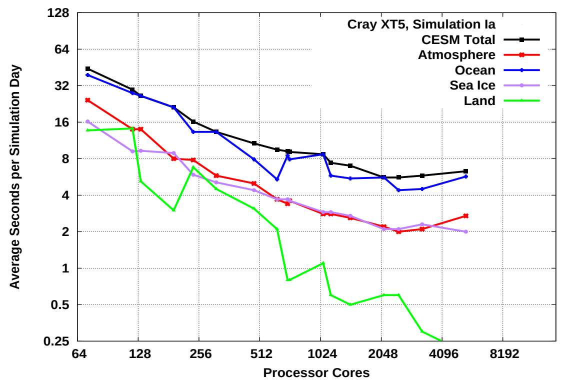


Figure 6: Component timing for best observed CESM performance for simulation Ia on the XT5

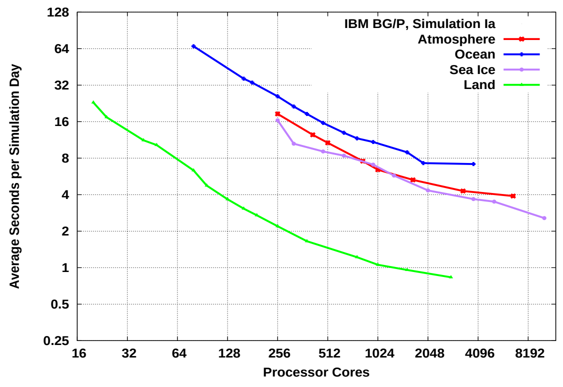


Figure 4: Best observed component performance for simulation Ia on the BG/P

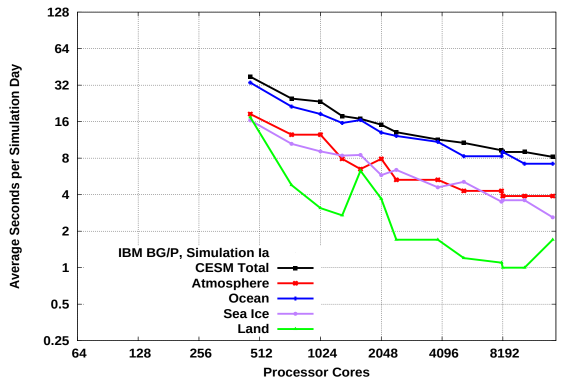


Figure 7: Component timing for best observed CESM performance for simulation Ia on the BG/P

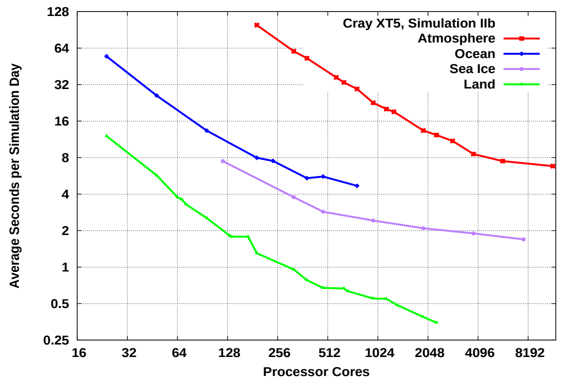


Figure 5: Best observed component performance for simulation IIb on the XT5

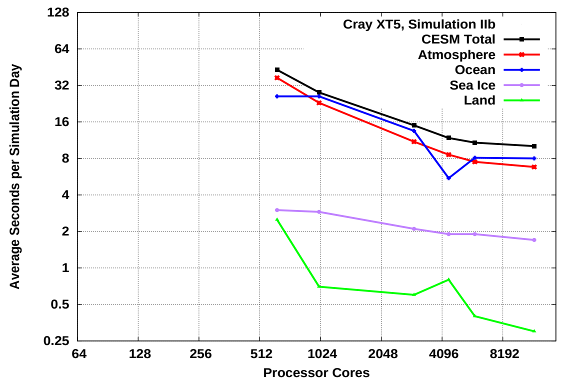


Figure 8: Component timing for best observed CESM performance for simulation IIb on XT5

component for the CESM performance data for simulation Ia in Fig. 2, respectively. Qualitatively there is little apparent difference between these results and those on the XT5. The curves simply appear to be shifted to the right. In particular, good configurations still balance ocean cost with that of the atmosphere, sea ice, and coupler. The fact that the atmosphere and sea ice does not increase in cost at the largest processor core counts allows performance to continue scaling on the BG/P for this problem. For the CESM configuration using 8,192 total cores, atmosphere used 6,656 (1664 × 4), sea ice used 5,120 (1280 × 4), and ocean used 1,536 (385 × 4). For the largest configuration, using 14,464 cores, atmosphere was unchanged but sea ice now used 10,240 (2560 × 4) and ocean used 3,840 (960 × 4).

In another comparison, Figures 5 and 8 are component performance scaling for problem IIb on the XT5 and time spent in each component for the CESM performance data for simulation IIb in Fig. 1, respectively. Here the atmosphere is the most expensive component, as would be expected from the larger grid and the 7 times more expensive atmospheric physics. For the largest configuration examined, using a total of 11,712 processor cores, atmosphere used 11,520 (1920 × 6), sea ice used 9,216 (1536 × 6), and ocean used 192 (32 × 6). Overall CESM performance for simulation IVb was determined largely by that of the atmosphere and sea ice components.

6. HIGH RESOLUTION SIMULATIONS

We examine performance for 2 large simulations. The first represents the finest resolution under consideration for use with the FV dynamical core.

III) 0.23 × 0.31 degree resolution horizontal grid (1152 × 768)

This is coupled with ocean and sea ice using a 0.1 degree resolution horizontal computational grid (2400 × 3600) and 42 vertical levels for the ocean. We consider only B1850_CN, and refer to this problem as IIIa. We also have data only from the XT5, and only an approximate optimization of the CESM configurations was attempted.

Figure 9 shows the component performance scaling for problem IIIa on the XT5. Figure 10 shows the time spent in each component for the best observed CESM performance for a given total number of processor cores. For the maximum configuration (31,488 processor cores, achieving a throughput rate of 2.7 simulated years per day) the atmosphere used 19,968 cores (3328×6), sea ice used 21,600 (3600×6, the first 19,968 shared with the atmosphere) and ocean used 9,120 (1520 × 6). Performance scalability is an issue for all of the components except the land at these fine resolutions. Significant efforts are being applied to improve the performance of all components, and alternative, more scalable, numerical methods are being investigated. The atmosphere is the first component for which a scientifically acceptable alternative approach has been developed.

Figure 11 is a graph of throughput rates for CAM when using the FV dycore and when using a spectral element dycore (SE) configured to run on a cubed-sphere grid. Unlike the longitude-latitude grid used by the FV dycore, the cubed-sphere grid does not cluster grid points at the poles. This

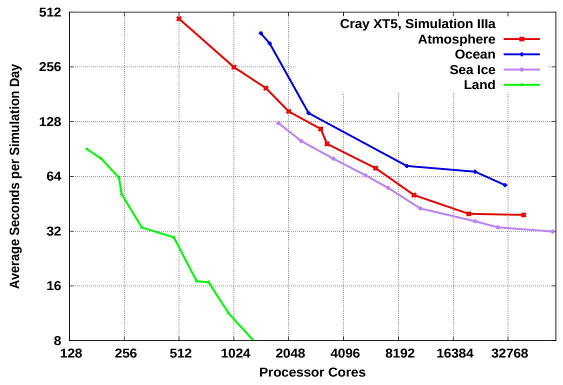


Figure 9: Best observed component performance for simulation IIIa on the XT5

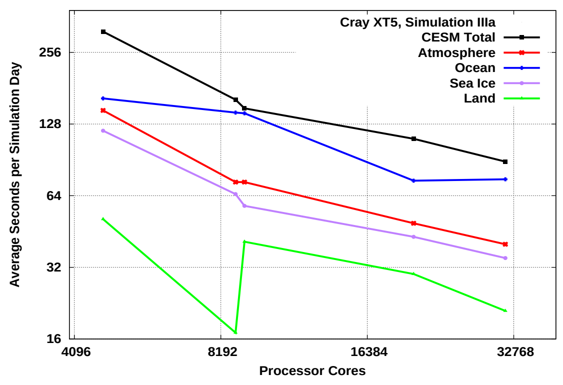


Figure 10: Component timing for best observed CESM performance for simulation IIIa on the XT5

allows for a number of numerical and parallel algorithm simplifications that improve performance scaling. For example, the parallel implementation of the SE dycore uses only a single decomposition of the horizontal extent of the computational grid. The data plotted in Fig. 11 are from a standalone test of CAM that is similar, but not identical, to the atmosphere component scalings presented in the previous figures. The SE grid ne120np4 has 777,602 grid points, as compared to 884,736 for the 0.23 × 0.31 FV grid, and is considered comparable in terms of the resulting solution accuracy. As can be seen, the SE dycore performance and performance scaling is much better than that of the FV dycore. The simpler communication requirements also maps very well to the BG/P architecture, and CAM using the SE dycore on the BG/P is as fast as on the XT5 at scale. This is without the more expensive CAM5 physics, however.

Our final large simulation uses the SE dycore and an even larger horizontal grid for the atmosphere:

IV) ne240np4 horizontal grid (3, 110, 402 gridpoints with an approximately 0.125 degree resolution)

This is coupled with the same 0.1 degree ocean/sea ice grid used before. We again consider only B1850_CN, and refer to this problem as IVa. For this problem we have performance data for a selection of processor core counts up to nearly the full XT5 system size. Throughput rates for these runs are

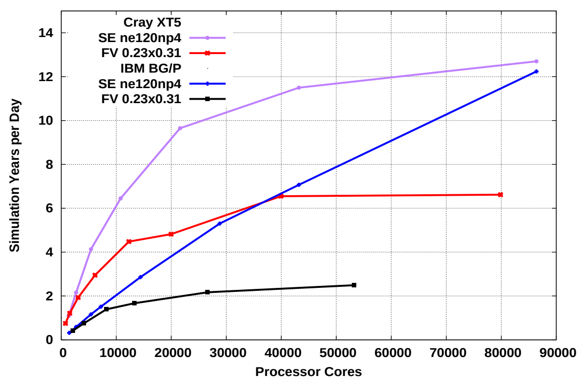


Figure 11: Best observed tmosphere performance using FV and SE dycores for simulation IIIa on the XT5 and BG/P

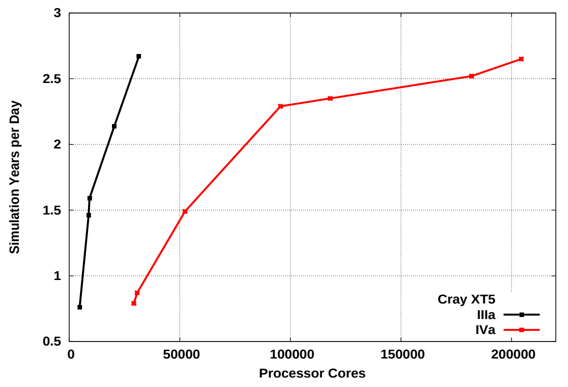


Figure 12: Best observed CESM performance for simulations IIIa, IVa on the XT5

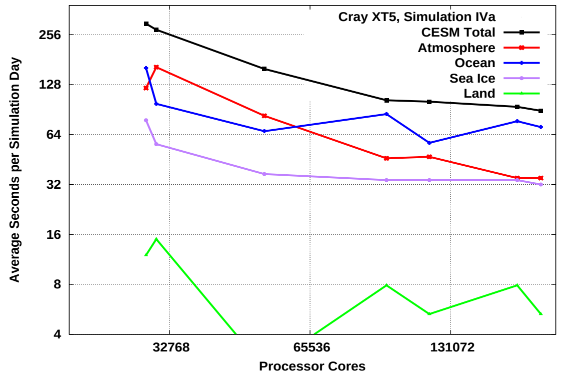


Figure 13: Component timing for best observed CESM performance for simulation IVa on the XT5

plotted in Figure 12. For comparison, the throughput rates for simulation IIIa are also included.

While there is still work to do, solving problem IVa (with its 4 times larger horizontal grid) with the SE dycore took the same amount of time as solving problem IIIa with FV. For 95,518 cores, the simulation achieved a rate of 2.3 simulated years per day, using 86,400 cores in the atmosphere (14400×6), 28,800 in sea ice (4800×6), and 9,120 in ocean (5120×6). For the largest configuration (204,407 cores, achieving a throughput rate of 2.7 simulated years per day), the atmosphere now used 172,800 cores (14400×12), sea ice used 57,600 (4800×12), and ocean used 31,608 (2634×12).

The per component timing for the IVa CESM run is described in Figure 13. From these experiments we have determined that, in addition to ocean and sea ice, we need to start looking seriously at additional performance optimization in the coupler.

7. CONCLUSIONS

CESM supports a rich assortment of performance tuning options that allow custom optimizations for different science options, grid sizes, processor counts and target platforms. Of particular utility compared to the predecessor code CCSM3 is the ability for some components to share processors and others to run on disjoint processor subsets. On the XT5 and BG/P, performance and performance scalability are acceptable even for relatively small problems, and the ability to increase the cost of the physics from current levels without seriously impacting parallel performance at scale will accelerate research into improved physical process modeling.

For high resolution studies, significant performance scaling problems remain, but new numerical methods are being developed that show great promise in improving performance at scale. The spectral element dycore in particular addresses many of the concerns about the atmosphere component. Similar alternative algorithm work is now targeting the ocean and sea ice. In the meantime, efforts to further optimize performance of the existing ocean and sea ice components continue.

8. ACKNOWLEDGMENTS

The work of A. Mirin and P. Worley was sponsored in part by the Climate and Environmental Sciences Division of the Office of Biological and Environmental Research and by the Office of Advanced Scientific Computing Research, both in the Office of Science, U.S. Department of Energy, under Contract No. DE-AC52-07NA27344 with Lawrence Livermore National Security, LLC. and Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC, respectively. The work of M. Taylor was supported by SciDAC grant 06-13194 from the Office of Biological and Environmental Research, U.S. Department of Energy. The work of A. Craig and M. Vertenstein was supported by the Office of Biological and Environmental Research, U.S. Department of Energy under contracts DE-FC02-97ER62402 and DE-FC02-07ER64340. Additional support for M. Vertenstein was provided by National Science Foundation grant AGS-0856145. This work used resources of the Oak Ridge Leadership Computing Facility, located in the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725. It also used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

9. REFERENCES

[1] C. M. Bitz and W. H. Lipscomb. An energy-conserving thermodynamic model of sea ice. *Journal of Geophysical Research*, 104:15669–15677, 1999.

[2] W. D. Collins, C. M. Bitz, et al. The Community Climate System Model Version 3 (CCSM3). *J. Climate*, 19(11):2122–2143, 2006.

[3] W. D. Collins, P. J. Rasch, et al. The Formulation and Atmospheric Simulation of the Community Atmosphere Model: CAM3. *Journal of Climate*, 19(11):2144–2161, June 2006.

[4] Community Earth System Model. http://www.cesm.ucar.edu/.

[5] A. P. Craig, M. Vertenstein, and R. Jacob. A new flexible coupler for earth system modeling developed for CCSM4 and CESM1. *Int. J. High Perf. Comput. Appl.*, 2011. (accepted).

[6] J. M. Dennis, J. Edwards, et al. An application level parallel i/o library for earth system models. *Int. J. High Perf. Comput. Appl.*, 2011. (accepted).

[7] J. M. Dennis, J. Edwards, et al. CAM-SE: A scalable spectral element dynamical core for the community atmosphere model. *Int. J. High Perf. Comput. Appl.*, 2011. (under review).

[8] R. E. Dickinson, K. W. Oleson, G. Bonan, F. Hoffman, P. Thornton, M. Vertenstein, Z.-L. Yang, and X. Zeng. The Community Land Model and its climate statistics as a component of the Climate System Model. *Journal of Climate*, 19(11):2032–2324, 2006.

[9] J. Drake, P. Jones, M. Vertenstein, J. White III, and P. Worley. Software design for petascale climate science. In D. Bader, editor, *Petascale Computing: Algorithms and Applications*, chapter 7, pages 125–146. Chapman & Hall/CRC, New York, NY, 2008.

[10] J. B. Drake, P. W. Jones, and G. Carr. Overview of the software design of the Community Climate System Model. *Int. J. High Perf. Comput. Appl.*, 19(3):177–186, Fall 2005.

[11] J. K. Dukowicz, R. D. Smith, and R. C. Malone. Implicit free-surface method for the Bryan-Cox-Semtner ocean model. *J. Geophys. Res.*, 99:7991–8014, 1994.

[12] P. R. Gent, G. Danabasoglu, et al. The community climate system model version 4. *Journal of Climate*, 2011. (accepted).

[13] F. Hoffman, M. Vertenstein, P. Thornton, K. Oleson, and S. Levis. Community Land Model version 3.0 (CLM3.0) developer's guide. Technical Report ORNL/TM-2004/119, Oak Ridge National Laboratory, Oak Ridge, TN, June 2004.

[14] F. M. Hoffman, M. Vertenstein, H. Kitabata, J. B. White III, P. H. Worley, and J. B. Drake. Adventures in vectorizing the Community Land Model. In R. Winget and K. Winget, editor, *Proceedings of the 46th Cray User Group Conference, May 17-21, 2004*, Eagan, MN, 2004. Cray User Group, Inc.

[15] E. C. Hunke and J. K. Dukowicz. An elastic-viscous-plastic model for sea ice dynamics. *J. Phys. Oceanogr.*, 27:1849–1867, 1997.

[16] E. C. Hunke and W. H. Lipscomb. CICE: the Los Alamos Sea Ice Model documentation and software user's manual version 4.1. Technical Report LA-CC-06-012, Los Alamos National Laboratory, Los Alamos, NM, May 2010.

[17] P. W. Jones, P. H. Worley, Y. Yoshida, J. B. White III, and J. Levesque. Practical performance portability in the Parallel Ocean Program (POP). *Concurrency and Computation: Practice and Experience*, 17(10):1317–1327, 2005.

[18] S.-J. Lin. A 'vertically Lagrangian' finite-volume dynamical core for global models. *Mon. Wea. Rev.*, 132(10):2293–2307, 2004.

[19] A. Mirin and W. B. Sawyer. A scalable implemenation of a finite-volume dynamical core in the Community Atmosphere Model. *Int. J. High Perf. Comput. Appl.*, 19(3):203–212, Fall 2005.

[20] A. A. Mirin and P. H. Worley. Improving the performance scalability of the community atmosphere model. *Int. J. High Perf. Comput. Appl.*, 2011. (accepted).

[21] R. B. Neale, C.-C. Chen, et al. Description of the NCAR Community Atmosphere Model (CAM 5.0). NCAR Tech Note NCAR/TN-???+STR, National Center for Atmospheric Research, Boulder, CO 80307, June 2010.

[22] Office of Science, U.S. Department of Energy. A Science-Based Case for Large-Scale Simulation. (available from http://www.pnl.gov/scales/), July 30 2003.

[23] K. W. Oleson, D. M. Lawrence, et al. Technical description of version 4.0 of the Community Land Model (CLM). NCAR Tech Note NCAR/TN-478+STR, National Center for Atmospheric Research, Boulder, CO 80307, April 2010.

[24] I. C. Rutt, M. Hagdorn, N. J. R. Hulton, and A. J. Payne. The Glimmer community ice sheet model. *Journal of Geophysical Research*, 114:1–22, April 2009. F02004.

[25] R. D. Smith, P. W. Jones, et al. The Parallel Ocean Program (POP) reference manual: Ocean component of the Community Climate System Model (CCSM). Technical Report LAUR-10-01853, Los Alamos National Laboratory, Los Alamos, NM, March 2010.

[26] P. H. Worley and J. B. Drake. Performance portability in the physical parameterizations of the Community Atmosphere Model. *International Journal of High Performance Computing Applications*, 19(3):187–202, August 2005.